CLAIMS

What is claimed is:

1. A method for generating network topology information, the method comprising:

identifying a first link state packet associated with a first node;

identifying a second link state packet associated with a second node;

generating network topology information associated with the first node using information from the first link state packet; and

deferring verification of two-way connectivity checking between the first node and the second node until analysis of second link state packet is initiated for generating network topology information associated with first second node.

2. The method of claim 1, wherein the second link state packet is not analyzed solely for purposes of verifying two-way connectivity.

3. The method of claim 1, further comprising:

setting a two-way connectivity flag in response to determining that two-way connectivity exists between first and second nodes, wherein the two-way connectivity flag indicates that a valid two-way connection exists between first and second nodes.

4. The method of claim 1, wherein a PATH data structure and a TENT data structure are maintained.

5. The method of claim 1, wherein the first link state packet is scanned only once.

6. The method of claim 1, wherein the second link state packet is scanned only once.

7. The method of claim 1, wherein the network topology is generated using Intermediate System-Intermediate System Link State Routing.

8. The method of claim 1, wherein the network topology is generated using Open Shortest Path First Routing.

9. The method of claim 1, further comprising generating routing table information using the network topology information.

10. The method of claim 4, wherein the PATH and TENT data structures are rolled-back in response to a determination that two-way link connectivity checking has failed.

11. The method of claim 1, wherein reverse path insertion is used to generate network topology information in response to a determination that two-way link connectivity checking has failed.

12. The method of claim 4, wherein the PATH data structure is associated with a verification flag used for two-way connectivity checking.

13. The method of claim 4, wherein the PATH data structure is associated with multiple verification flags used for two-way connectivity checking, wherein the number of verification flags is equal to the maximum number of parent nodes.

14. The method of claim 13, wherein a first verification flag is set to true when two-way connectivity is verified between a first node and a second node.

15. The method of claim 14, wherein a second verification flag is set to true when two-way connectivity is verified between a first node and a third node.

16. A method for generating network topology information, the method comprising:

receiving link state packets from a plurality of network nodes;

performing a single scan of the link state packets from the plurality of network nodes to generate network topology information associated with the plurality of network nodes, wherein the single scan performs two-way connectivity checking between two or more network nodes.

17. The method of claim 16, further comprising:

generating routing table information using the generated topology information.

18. A method for generating a network topology, the method comprising:

generating first network topology information using a first link state packet from a first node; and

generating second network topology information using a second link state packet from a second node, wherein generating second network topology information comprises performing two-way connectivity verification between the first node and the second node.

19. The method of claim 18, wherein the second link state packet is scanned only once to generate the second network topology information.

20. A computer program product comprising a machine readable medium on which is provided program instructions for generating a network topology, the computer program product comprising:

computer code for generating first network topology information using a first link state packet from a first node; and

computer code for generating second network topology information using a second link state packet from a second node, wherein generating second network topology information comprises performing two-way connectivity verification between the first node and the second node.

21. The computer program product of claim 20, wherein the second link state packet is not analyzed solely for purposes of verifying two-way connectivity.

22. The computer program product of claim 20, further comprising:

setting a two-way connectivity flag in response to determining that two-way connectivity exists between first and second nodes, wherein the two-way connectivity flag indicates that a valid two-way connection exists between first and second nodes.

23. The computer program product of claim 20, wherein logs of PATH and TENT data structures are maintained.

24. The computer program product of claim 20, wherein the first link state packet is scanned only once.

25. The computer program product of claim 20, wherein the second link state packet is scanned only once.

26. The computer program product of claim 20, wherein the network topology is generated using Intermediate System-Intermediate System Link State Routing.

27. The computer program product of claim 20, wherein the network topology is generated using Open Shortest Path First Routing.

28. The computer program product of claim 20, wherein network topology information is generated using Dijkstra's algorithm.

29. The method of claim 20, further comprising generating routing table information using the network topology information.

30. The method of claim 20, wherein two-way link connectivity checking fails.

31. The method of claim 30, wherein reverse path insertion is used where two-way link connectivity checking fails.

32. An apparatus for generating network topology information, the apparatus comprising:

memory;

at least one processor coupled with memory; and

at least one interface;

the processor being configured to generate first network topology information using a first link state packet from a first node and generate second network topology information using a second link state packet from a second node, wherein generating second network topology information comprises performing two-way connectivity verification between the first node and the second node.

33. The apparatus of claim 32, wherein the processor is further configured to scan a second link state packet once to generate the second network topology information.

34. The apparatus of claim 32, wherein the processor is further configured to set a two-way connectivity flag in response to determining that two-way connectivity exists between first and second nodes, wherein the two-way connectivity flag indicates that a valid two-way connection exists between first and second nodes.

35. The apparatus of claim 32, wherein the processor is further configured to stored into memory a PATH and a TENT data structure.

36. The apparatus of claim 35, wherein the PATH data structure is associated with at least one verification flag used for verifying two-way connectivity checking.

37. The method of claim 35, wherein the PATH data structure is associated with multiple verification flags used for two-way connectivity checking, wherein the number of verification flags is equal to the maximum number of parent nodes.

38. The method of claim 37, wherein a first verification flag is set to true when two way connectivity is verified between a first node and a second node.

39. The method of claim 38, wherein a second verification flag is set to true when two way connectivity is verified between a first node and a third node.

40. The apparatus of claim 32, wherein the processor is configured to scan the second link state packet once.

41. The apparatus of claim 32, wherein the processor is configured to generate network topology using Intermediate System-Intermediate System Link State Routing.

42. The apparatus of claim 32, wherein the processor is configured to generate network topology information using Dijkstra's algorithm.

43. The apparatus of claim 32, wherein the processor is configured to generate routing table information using the network topology information.

44. The apparatus of claim 35, wherein the processor is configured to roll-back the PATH and TENT data structures in response to a determination that two-way link connectivity checking has failed.

45. The apparatus of claim 32, wherein the processor is configured to use reverse path insertion to generate network topology information in response to a determination that two-way link connectivity checking has failed.

46. An apparatus for generating network topology information, the apparatus comprising:

means for identifying a first link state packet associated with a first node;

means for identifying a second link state packet associated with a second node;

means for generating network topology information associated with the first node using information from the first link state packet; and

means for deferring verification of two-way connectivity checking between the first node and the second node until analysis of second link state packet is initiated for generating network topology information associated with second node.

47. The apparatus of claim 46, wherein the second link state packet is not analyzed solely for purposes of verifying two-way connectivity.

48. The apparatus of claim 46, further comprising:

means for setting a two-way connectivity flag in response to determining that two-way connectivity exists between first and second nodes, wherein the two-way connectivity flag indicates that a valid two-way connection exists between first and second nodes.

49. The apparatus of claim 46, wherein logs of PATH and TENT data structures are maintained.

50. The apparatus of claim 46, wherein the first link state packet is scanned only once.

51. The apparatus of claim 46, wherein the second link state packet is scanned only once.

52. The apparatus of claim 46, wherein the network topology is generated using Intermediate System-Intermediate System Link State Routing.

53. The apparatus of claim 46, wherein network topology information is generating using Dijkstra's algorithm.

54. The method of claim 46, further comprising generating routing table information using the network topology information.

55. The method of claim 46, wherein two-way link connectivity checking fails.

56. The method of claim 55, wherein reverse path insertion is used where two-way link connectivity checking fails.